UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

telecom BCN

Towards Machine Learning applied to Time Series based Network Traffic Forecasting

Thesis Submitted to the Faculty of the Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by Javier González Prieto

In partial fulfilment of the requirements for the degree in

TELEMATIC ENGINEERING

Advisor: Albert Cabellos Aparicio

Barcelona, September 2016

Abstract

The aim of this project is to implement a network traffic forecasting model using time series and improve its performance with machine learning techniques, offering a better prediction based in outlier correction. This is a project developed in the Computer Architecture Department (DAC) at the Universitat Politècnica de Catalunya (UPC).

Time Series modeling methodology is able to shape a trend and take care of any existing outlier, however it does not cover outlier impact on forecasting. In order to achieve more precision and better confidence intervals, the model combines outlier detection methodology and Artificial Neural Networks to quantify and predict outliers. A study is realized over external data to find out if there is an improvement and its effect on the predictions.

Machine learning techniques as Artificial Neural Networks has proven to be an improvement of the current methodology to realize forecasting using Time Series modeling. Future work will be oriented to create an improved standard of this system focused on generalize the model.

Acknowledgments

I wish to express my sincere thanks to Dr. Albert Cabellos, Professor of the Computer Architecture Department at the Universitat Politècnica de Catalunya, for giving me the opportunity to work with him and trust in me during the last stage of my degree.

I place on record, my sincere thanks to Albert Mestres, PhD candidate of the Computer Architecture Department at Universitat Politècnica de Catalunya, for helping to accomplish my goal and give me his unconditional support.

I also want to acknowledge Josep A. Sànchez Espigares, Professor at the Interuniversity Master's Degree in Statistics for collaborating with the project and provide us with his knowledge.

Thanks to my friends, for being all these years beside me and for loving me as I am.

Finally, thanks to my family, specially Andrés González, my father; Lucia Prieto, my mother; and David González, my brother who have always been there to give me the support I needed and for all the work done to bring me up and allow me to be where I am.

Revision, history and approval record

Revision	Date	Purpose
0	06/09/2016	Document creation
1	18/09/2016	Document revision
2	22/09/2016	Document revision

Document distribution list

Name	e-mail
Javier González Prieto	javigonzpriet@gmail.com
Albert Cabellos Aparicio	albert.cabellos@gmail.com

Written by:		Reviewed and approved by:	
Date	06/09/2016	Date	22/09/2016
Name	Javier González Prieto	Name	Albert Cabellos Aparicio
Position	Project Author	Position	Project Supervisor

Table of contents

List of figures

List of tables

1. Introduction

The project has been developed at the Computer Architecture Department (CAD) at the Universitat Politècnica de Catalunya (UPC). The project aims to improve a specific Time Series modeling forecasting (ARIMA) with Machine Learning techniques and outlier detection.

1.1. Statement of purpose

This project deals with the implementation of a Time Series Model (TSM) supported on a Neural Network (NN) to improve its efficiency working over Network data. The Time Series Model must be strong enough to provide a prediction within a specific deviation error. This means that the Network Data has a pattern over time and therefore can be predicted.

After the TSM is adjusted to the data, there can be atypical measurements that do not fit into any model since they are generated by punctual situations such as festivities, timetable changes and particular situations.

The main purpose of this project is to demonstrate that, using Machine Learning (ML) techniques, it is possible to correlate external data with a time series model to improve the predictions and adjustment by targeting the atypical measurements. Generating correlations between calendar data and network data it is possible to add corrections to the TSM and obtain a better forecast.

1.2. Requirements and specifications

Project requirements:

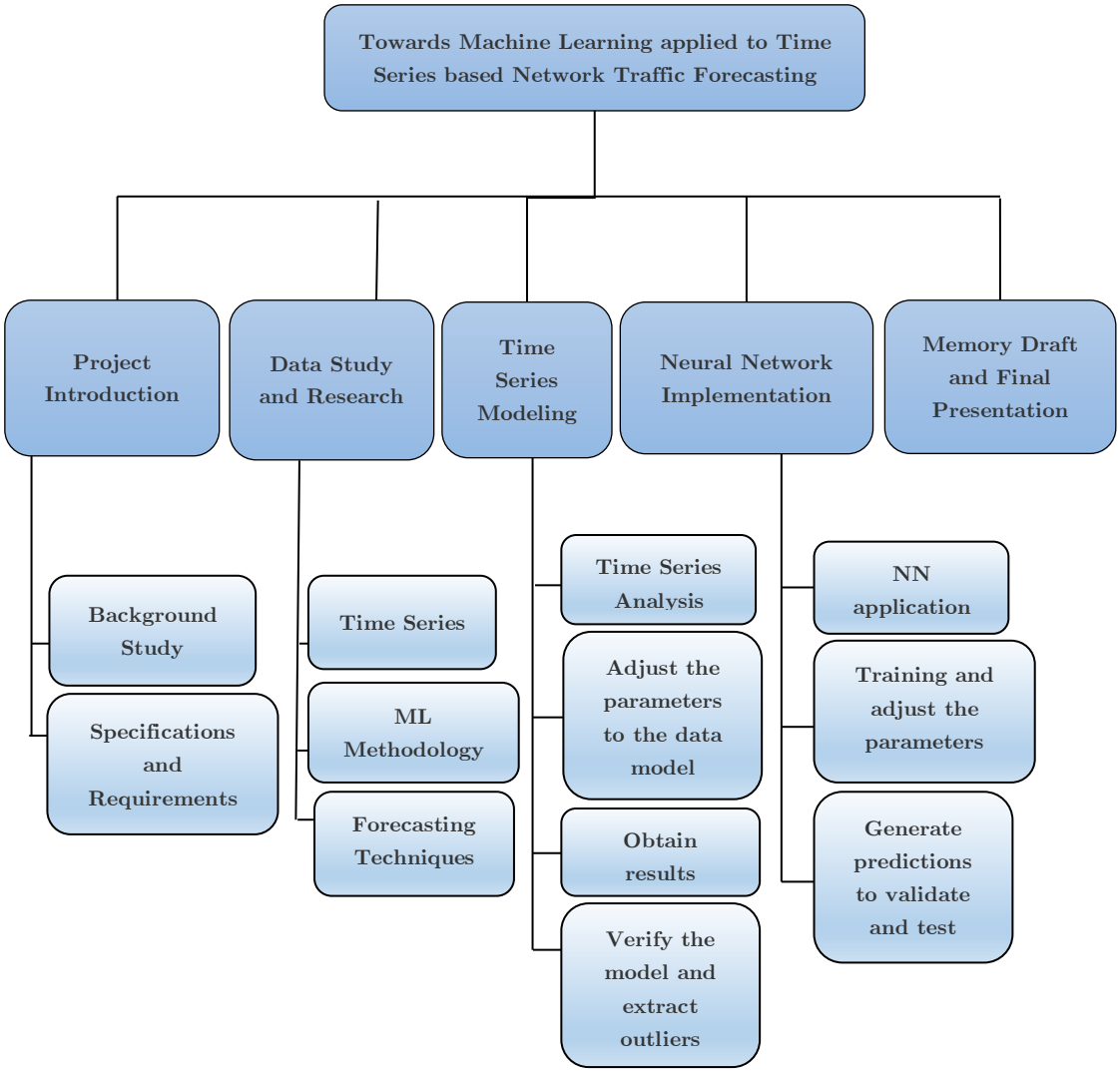
- Implement a Time Series Model adjustment fitted enough to guarantee an almost complete error correction without outliers.
- The use of an outlier measurement method precise enough to minimize the error of the TSM.
- Generate a NN structure stable and capable to reduce the deviation error produced by atypical situations.
- The NN must be able to correlate Calendar data to Network data in order to quantify and predict outliers in the future.

Project specifications:

- Minimum of 50 features of external data to correlate with the time series.
- Network data must be stable, seasonal and free of noise caused by network maintenance.
- Minimum of a calendar year of data.

1.3. Work packages, milestones and Gantt diagram

Work Breakdown Structure



Work packages

Project: Project Introduction	WP ref: 1	
Major constituent: Project definition	Sheet 6 of 11	
Short description: Setting the bases and objectives of the project and understand the requirements and specifications before getting started.	Planned start date: 22/02/2016 Planned end date: 11/03/2016	
	Start event: Meeting End event: ML knowledge	
Internal task T1: Background Study Internal task T2: Project specifications and requirements definition	Deliverables: Results	Dates:

Project: Data study and research	WP ref: 2	
Major constituent: Information sources research	Sheet 7 of 11	
Short description: Finding a way to use Deep Learning and Time Series modeling techniques with a particular database.	Planned start date: 14/03/2016 Planned end date: 27/05/2016	
	Start event: End event:	
Internal task T1: Database Research Internal task T2: ML techniques simulation	Deliverables: Knowledge	Dates:

<table>
<tr><td>Project: Time Series Modeling</td><td colspan="2">WP ref: 3</td></tr>
<tr><td>Major constituent: Testing and simulations</td><td colspan="2">Sheet 7 of 11</td></tr>
<tr><td rowspan="2">Short description:

Develop a TS to model the database, generate predictions and extract outliers.</td><td colspan="2">Planned start date: 30/05/2016

Planned end date: 28/07/2016</td></tr>
<tr><td colspan="2">Start event: Developing TS model

End event: Validating the results</td></tr>
<tr><td>Internal task T1: Time Series Analysis

Internal task T2: Adjust the parameters

Internal task T3: Obtain results to predict the best performance model

Internal task T4: Verify the model and extract outliers</td><td>Deliverables:

Simulations

Results

Predictions

Statistics</td><td>Dates:</td></tr>
</table>

<table>
<tr><td>Project: Neural Network Implementation</td><td>WP ref: 4</td></tr>
<tr><td>Major constituent: Testing and simulations</td><td>Sheet 8 of 11</td></tr>
<tr><td rowspan="2">Short description:

Develop a Neural Network structure that fits the outliers and extract the results.</td><td>Planned start date: 29/07/2016

Planned end date: 05/09/2016</td></tr>
<tr><td>Start event: Data Validation

End event: Confirmation of the results</td></tr>
</table>

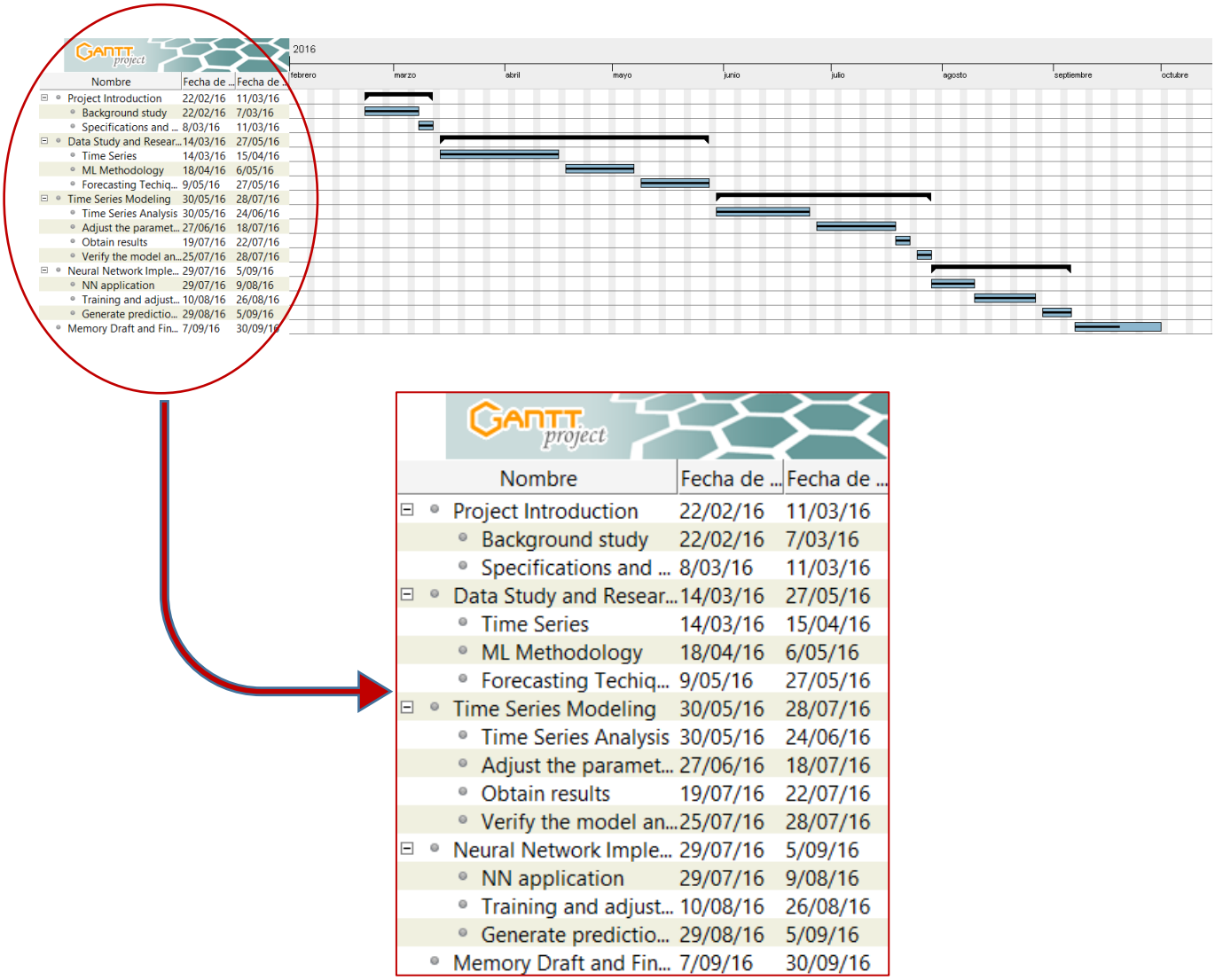
Internal task T1: NN application generation Internal task T2: Training and adjust the parameters Internal task T3: Generate predictions to validate and test	Deliverables: Results Draft	Dates:

Project: Memory Draft and Final Presentation	WP ref: 5	
Major constituent: Redacting the project final deliverable	Sheet 8 of 11	
Short description: Writing of the final memory and prepare the presentation	Planned start date: 07/09/2016 Planned end date: 30/09/2016	
	Start event: Project Draft End event: Final Project	
	Deliverables: Final Document	Dates:

Milestones

WP#	Task#	Short title	Milestone / deliverable	Date (week)
1	1	Background Study		07/03/2016
1	2	Project specifications and requirements definition	Results of the specifications and requirements	11/03/2016
2	1	Time Series Research	Knowledge	27/05/2016
2	2	ML Methodology	State of Art	15/04/2016
2	3	Forecasting Techniques	Implementation scheme	06/05/2016
3	1	Time Series Analysis	Results	27/05/2016
3	2	Adjust the parameters	Simulation	28/07/2016
3	3	Obtain best performance model	Results	24/06/2016
3	4	Verify the model and extract outliers	Outliers	18/07/2016
4	1	NN application	NN structure	22/07/2016
4	2	Training and adjust the parameters	Simulation	28/07/2016
4	3	Generate predictions to validate and test	Results	05/09/2016

Gantt Diagram



1.4. Deviation from the initial plan and incidences

There have been a couple of incidences during this project, that is the reason the schedule has been modified and the deadlines extended. In almost every project there are minor changes in the time schedule. The goal of the project and its methodology have slightly changed since the beginning. The initial objective was using only ML techniques to correlate Network data to Calendar data and generate predictions. This was discarded after some initial analysis of the data structure and seasonality. For it will produce very poor results with a significant level of ML techniques complexity.

After this first approach the decision was to follow the State of Art in time based forecasting and use TSM with an improvement based in ML techniques. This provides more stable results and combine two methods of prediction that have never been used together for this specific purpose.

The other incident that has caused delay in this project has been the difficulties to find adequate Network data to serve the purpose of the project. It needed to be of at least one calendar year, from one unique node of the net small enough no to generate random noise and seasonal. The first set of data used was from a hub in japan that gave service to half the country. The main problem with this dataset was that only registered 5 minutes of each day and that does not allow an acceptable fit within any model because of the random component it contained.

It took over a month to locate a new dataset that fit the project purpose and to complete the procedure to obtain it. This incident completely modified the established schedule and delayed the deadline of this project.

2. Background

In this part, the background needed to understand the project and to start its development is presented. It is divided in two parts, Time Series Analysis and Neural Networks.

2.1. Time Series Analysis

Time Series can be defined as a sorted collection of observations equally spaced in time, which can be discrete or continuous. Time series analysis is the study of times series data in order to extract relevant parameters or characteristics from it. These can be used to generate a mathematical model of the initial Time Series for forecasting implementation.

ARIMA models are statistical resources based on Autoregressive (AR), Integrated (I) and Moving Average (MA) mathematical models. These are commonly used among any kind of time series, with or without seasonality. Finding the model that best fits a Time Series is done by determining the degree used in the AR, I and MA models and adjusting the coefficients of the mathematical equation that generates the time series replica. In Appendix 1 there is more detailed information about ARIMA modeling techniques.

2.1.1. Outliers Detection

Atypical situations in a Time Series can cause outliers that generate a mismatch with the model. The three more common outliers are Additive Outliers (AO), Transitory Changes (TC) and Level Shifts (LS). The methods to determine these outliers were developed by Chang and Tiao (1983), Chang, Tiao and Chen (1988) and Chen and Liu (1993) respectively. They are based on a similar procedure, the transformation of the ARIMA model to an infinite AR model. The coefficients of the AR model are used to determine the exceptional residues that may lead to possible outliers. The irregularities that overcome a threshold are analyzed by a series of mathematical equations to determine which kind of outlier is the cause of the atypical value. These are explained more meticulously in Appendix 2.

2.1.2. Forecasting

The statistical forecast of samples within Time Series is realized by the transcription of the series model to an infinite AR model, which gives good results in the short term. The new values are generated by the autoregression to previous points and they continue the model structure. To obtain the confidence intervals of this prediction, the ARIMA model is transformed to an infinite MA model that makes possible the calculation of the variances because the different variables are independent between them. This enables a forecast accuracy up to 95%. Appendix 3 contains more information about this subject.

2.2. Neural Networks

Neural Networks (NN) are a highly versatile machine learning technique, very useful when processing pattern recognition and classification. A NN can be trained with a set of feature vectors to obtain a desired output. For this to be done it is needed to have a set of feature vectors each one with one target vector (the wanted output).

Each neuron has a weight in the inputs it receives. If it determines that these values are over a threshold, it sends another value to a group of neurons in the next layer and so on until they reach the output layer with the estimated target values, as shown in figure 1. These weights need to be adjusted in order to obtain the minimum error at the output.

These weights are modified using different methods, some of the most common are the following: Levenberg-Marquardt algorithm, Scaled Conjugate Gradient algorithm and Resilient Backpropagation algorithm.

In order to set an end point to the training phase and avoid overfitting the net, the main set of vectors is divided in 3 subsets: Training set, Validation set and Test set. Each time the weights are changed is called an epoch. For each epoch, the training algorithm runs over the training set of features and adjusts the weight. Then the net is tested with the validation set to assure the output error is decreasing. If it is, the previous process is repeated. Otherwise, it increases a variable of 'failed weight modifications'. Once the variable exceeds a threshold it stops the training. The test set is used to assure that the net has not reached overfitting and to evaluate if output errors are acceptable.

For this process to work properly there are some considerations to take into account. Different net structures have different performances. In order to obtain the best fitting to the dataset, it is necessary to adjust the number of hidden layers of neurons, the number of neurons for layer, the training algorithm to use, the function used to calculate the output error, etc. There is no standard methodology to set these parameters. It is a common approach to try them all and compare the results.

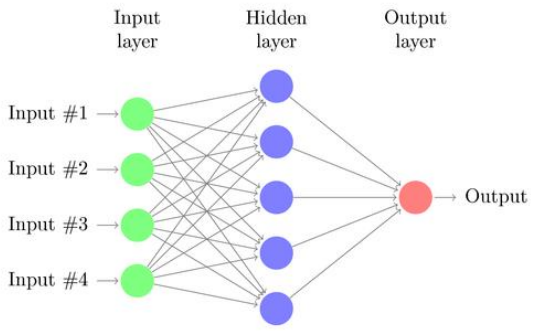


Figure 1: Neural Network scheme

3. Time Series Analysis

In the following section a study of the data used in the project is done in order to generate the best TSM and obtain the outliers for further treatment with machine learning techniques. The data has been granted by the Consorci de Serveis Universitaris de Catalunya (CSUC) and consists of different sets of data containing a daily hourly average of internet usage in the UPC Campus Nord. The dataset length goes from 20/06/2014 to 18/05/2016 and contains 699 samples.

3.1. Dataset Analysis

The fact that this dataset is from a University campus allows for detection of a strong weekly seasonality. On Saturdays and Sundays there is a considerable drop of the volume that can be detected just by plotting the sequence. In order to process seasonality correctly the dataset shall be reduced to 693 samples, starting on a Sunday and ending on a Saturday. The values are normalized to make them easier to analyze.

By plotting the weekly variances it can be appreciated that the volume passing through the node is increasing over time as the volume also grows.

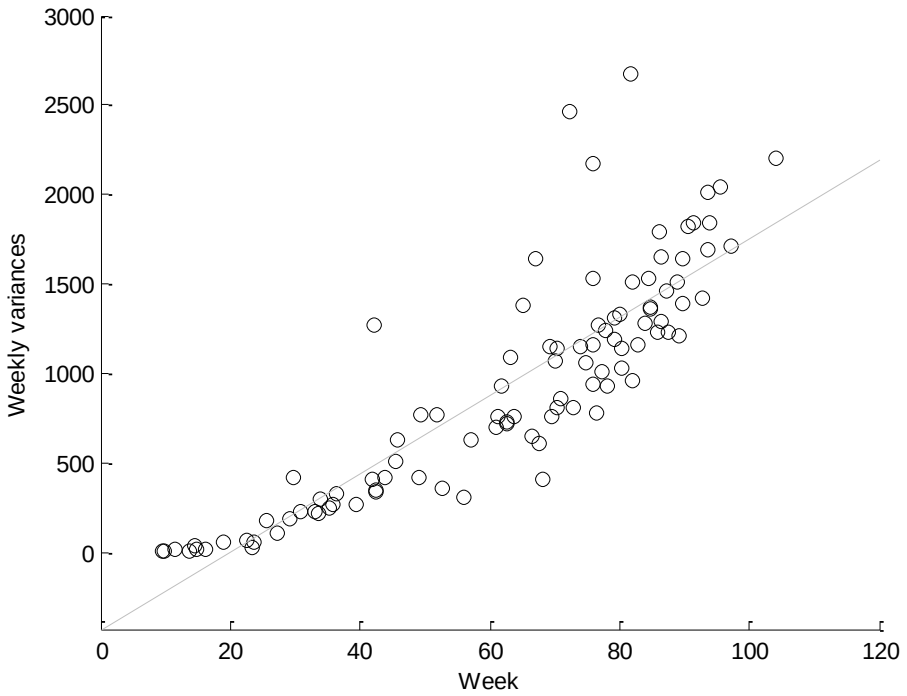


Figure 2: Weekly variances

The boxplot allows us to determine the weekly percentiles to detect the variations inside each week. A box plot is a way to analyze the variance of a set of samples, using the full range of variation (from min to max), the likely range of variation (the IQR) represented by the box height, and the mean value. The boxplot shown in figure 3 (left) is distributed in weeks and presents large boxes that means there is a great IQR within the week. In ARIMA modelling is important to reduce the variance of the TS if it is high. In our case, a logarithmic transformation is enough to correct this effect. In figure 3 (right) there is the representation of a boxplot of the logarithmic series and it can be appreciated that the difference between percentiles in a single week has been lowered. In the graph, this difference is represented by the length of the blue segments for each weekly data point. In the right figure a decrease of the length of the blue segments can be observed just looking at the left axis for both figures and noticing the difference in magnitude order, from hundreds to units.

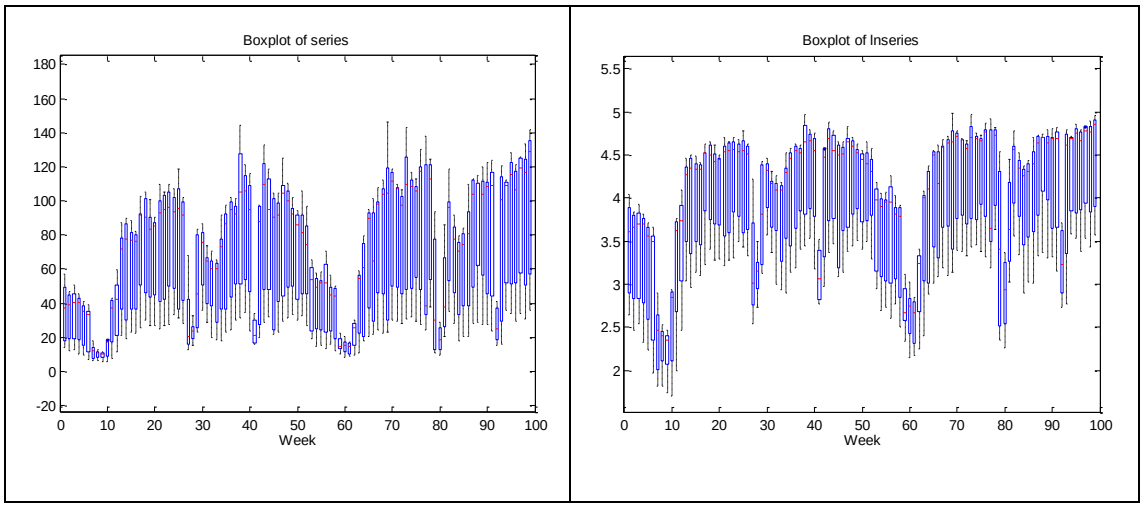


Figure 3: Series boxplot (left) and logarithmic series boxplot (right)

From now on the logarithmic series will be the one used in the analysis. To better observe the seasonal component, it is possible to decompose the TS in a trend, a seasonal component and a random component. The trend is obtained making the convolution between the series and a 7 length vector of value equal to 1/7. The seasonal component can be generated with the mean values of all the days of the week. Finally the random noise is the difference between the other two series.

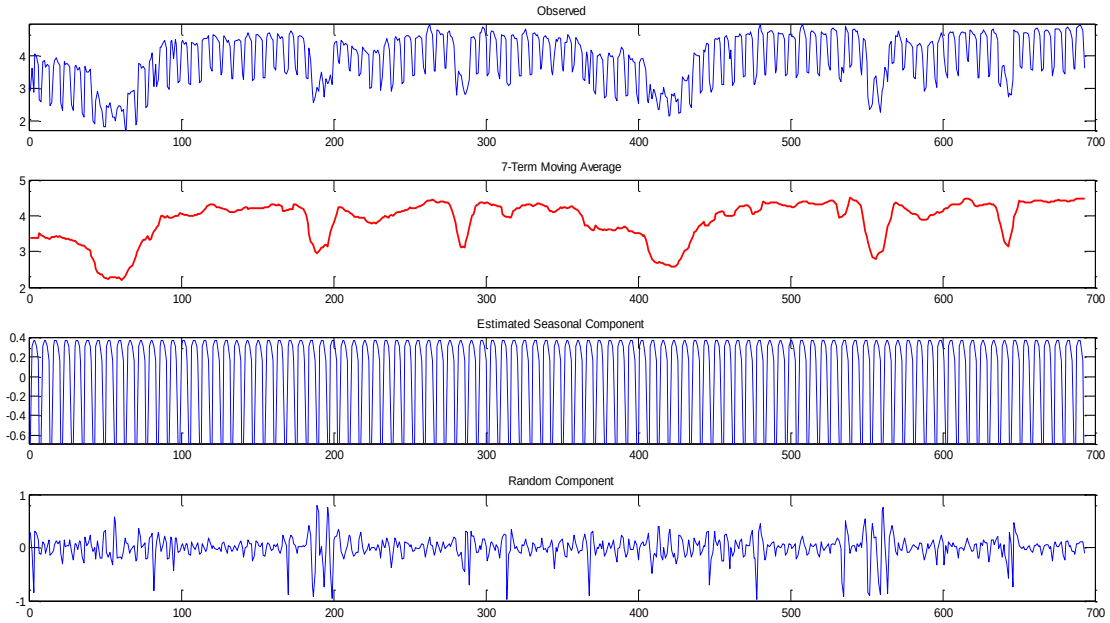


Figure 4: Time Series Decomposition

The subseries weekly plot represents additional confirmation that the seasonality is 7 days and has the expected behavior, (decreasing its volume during the weekend), as shown in figure 5. The Subseries plot is a representation of a period of a TS grouping the all the samples with the same position within the period, this case they correspond to the days of the week. Figure 5 presents the trend of all the Mondays, Tuesdays, etc. and the weekly trend composed by the mean values of each weekday.

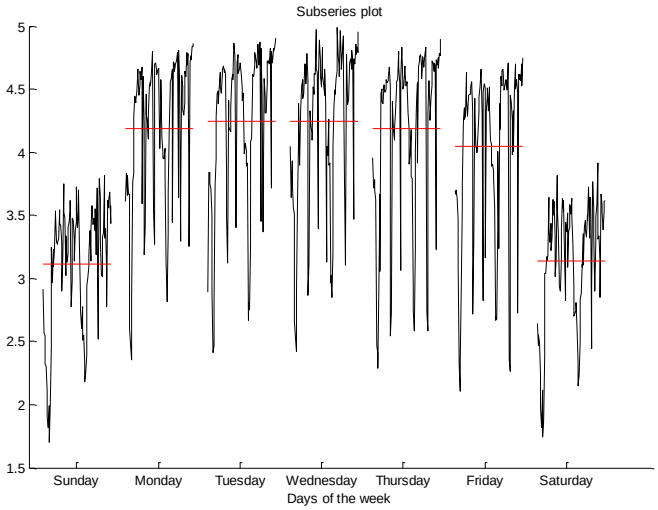


Figure 5: Weekly Subseries plot

Following Box-Jenkins methodology it is necessary to determine whether or not this series satisfy the requirements for ARIMA modeling, stationarity and mean value equal zero. Figure 6 contains the autocorrelation plot (left) and partial autocorrelation plot (right). The autocorrelation function (ACF) shows the correlation of samples by distance in the series. The partial autocorrelation function (PCF) is a similar concept to the previous one, taking into account the values in the intermediate positions. As show in figure 6, the autocorrelation function plot decreases slowly and that indicates non-stationarity. Weekly seasonality can be also appreciated each 7 lags.

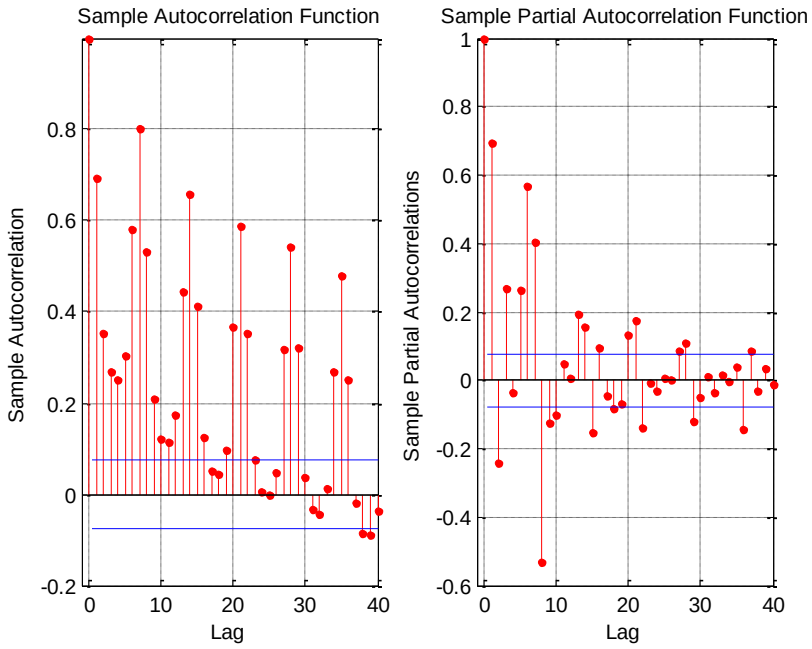


Figure 6: Autocorrelation and partial autocorrelation plot of the logarithmic series

Visualizing the logarithmic series, it is possible to detect that the mean value is not zero without any calculation. To solve this problem and the non-stationarity it is necessary to recourse to differentiation techniques. The differentiation process can also be done to address also the seasonality problem. Three possible differentiations are tested to determine the most adequate. An unique differentiation of 7 days, a double differentiation of 7 and one day and a triple differentiation of 7, one and one day. These three models are coded with d7lnseries for the lone differentiation, d7d1lnseries for the second one and d7d1d1lnseries for the last. The mean values and variances are shown in the following table. At first sight, d1d7lnseries is the best option because it is also stationary as can be detected in figure 7.

	Mean Value	Variance
d7lnseries	0.0114	0.2264
d1d7lnseries	5.0229e-04	0.1373
d1d1d7lnseries	-0.0011	0.3232

Table 1: Differentiation mean values and variances

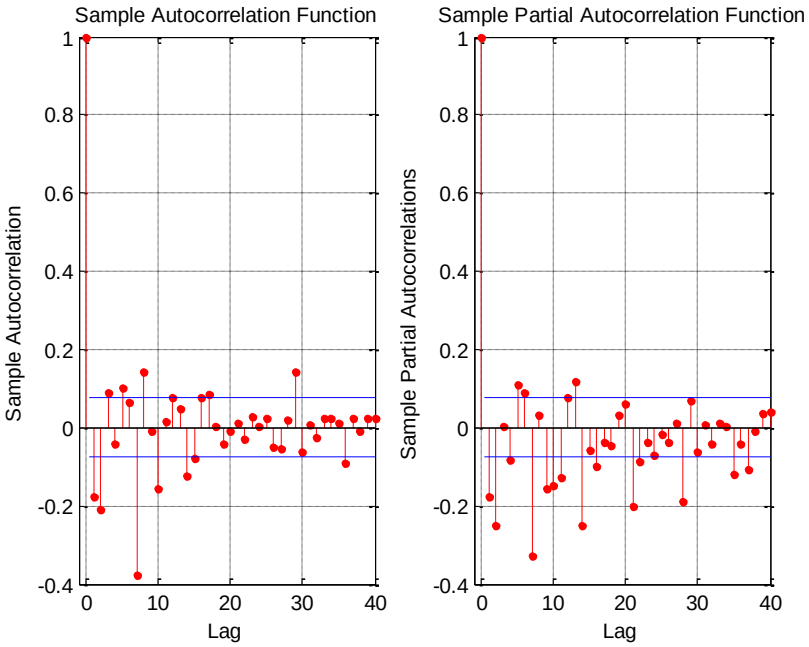


Figure 7: d1d7lnseries autocorrelation and partial autocorrelation functions

3.2. Model Identification

In this section there is a study to determine which ARIMA model fits better our time series. As ARIMA models are capable of integration and seasonal integration, the series does not have to be previously differentiated. The series to adjust is the logarithmic time series shown previously.

By analyzing the autocorrelation plot and partial autocorrelation plot (figure 7), it can be determined that an ARIMA(0,1,2)(0,1,2)7 model could fit the logarithmic series (lnseries). The highest autocorrelation values, apart from the ones caused by seasonal components, are in lags 1 and 2. That indicates a moving average of 2 lags. There is not an alternation of positive and negative values nor an exponential decay to zero, so there is no need for an autoregressive model. The seasonal components are look only by the multiples of 7 lags, there are only 2 of them over the threshold and there is no sign that points to an AR model. This means that are needed a MA(2) model with 1 days of differentiation and a MA(2) with 7 days of differentiation (seasonal).

The result is a model with MA coefficients -0.205645 and -0.266362 at lags 1 and 2, and SMA coefficients -0.677931 and -0.155755 at lags 7 and 14. The variance of the model is higher than expected: 0.0855336. The analysis of the residues of the model shows that there are outliers which do not fit into the model and produce deviation. If these are corrected the model would have much better performance.

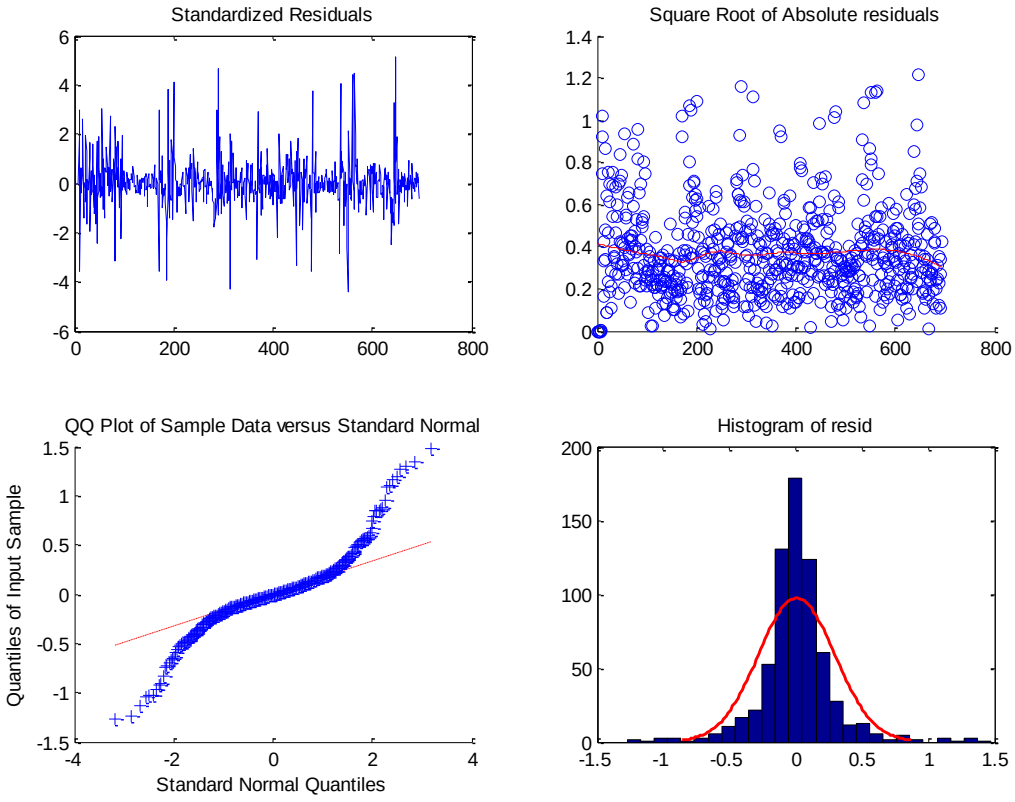


Figure 8: Residuals analysis

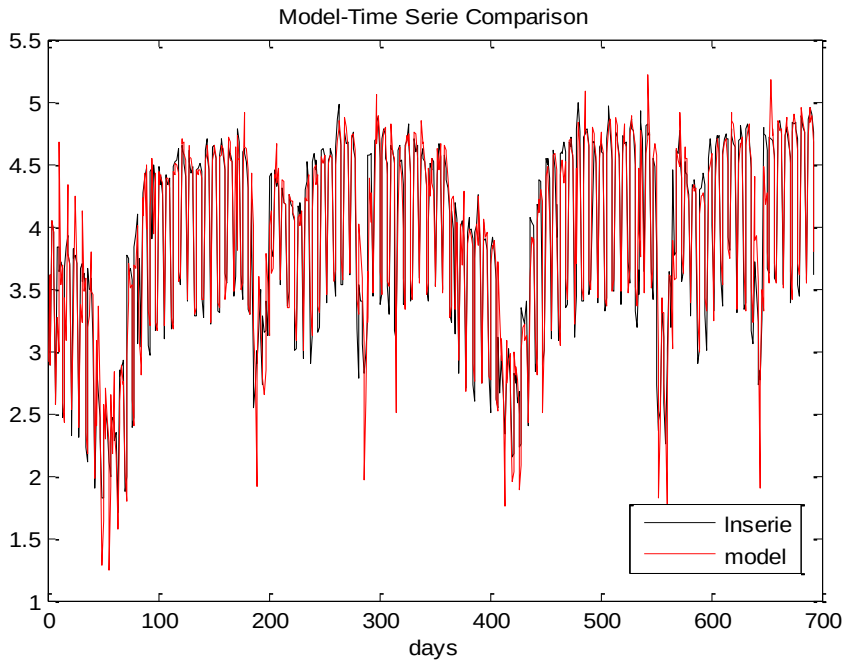


Figure 9: Comparison between logarithmic series and model

This model is aimed to do forecasting and is necessary to test how the outliers affect predictions. The model has been applied on all the sequence except to the last 3 weeks. This way it is possible to compare the results and determine the error. As shown in figure 10, the forecasting follows the trend of the series but the confidence intervals are too high and that indicates a larger possible error when making predictions. It is required to take care of the outliers in order to improve the performance.

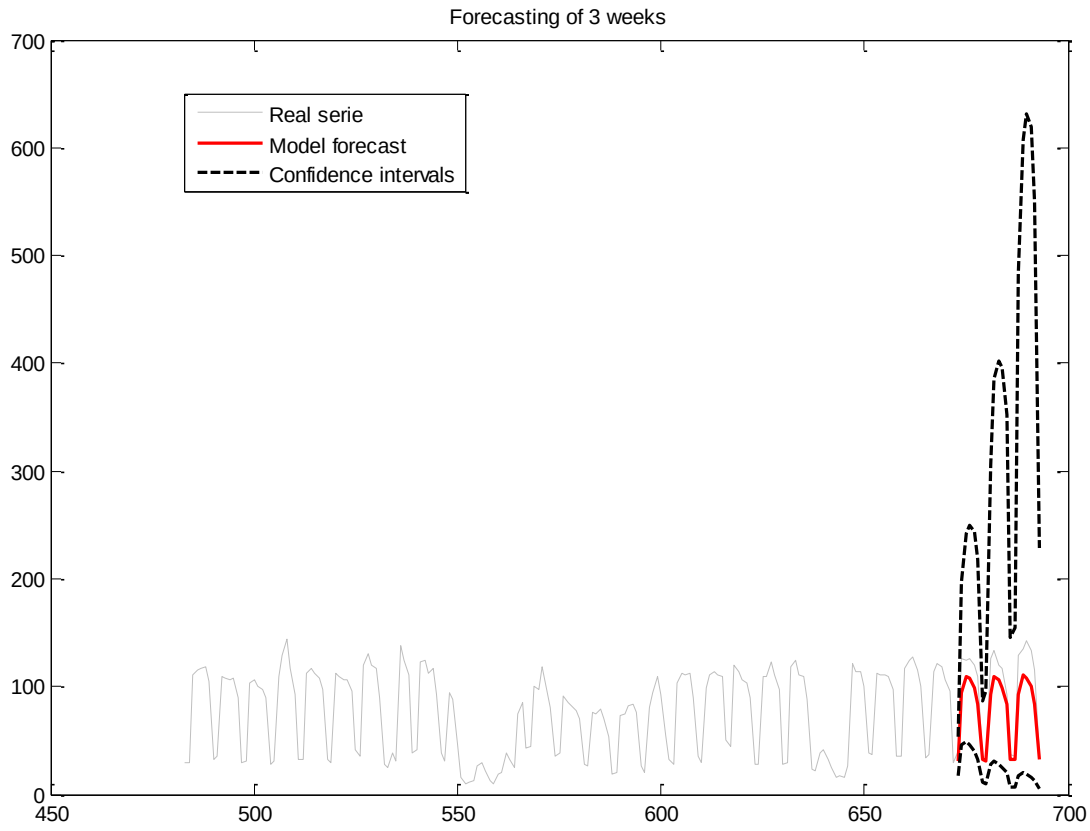


Figure 10: Forecasting of 3 weeks with outliers

3.3. Outlier detection

The outliers focused in this time series are TC and AO, due that LS is less common and has a more significant effect on the model. Following the methodology described in Appendix 2, the outliers overcoming a threshold of 2.6 on the residues are classified and quantified. In this set of samples 79 outliers have been found: 55 AO and 24 TC. If these are subtracted to the main time series, the model is better adjusted as can be appreciated in figure 11.

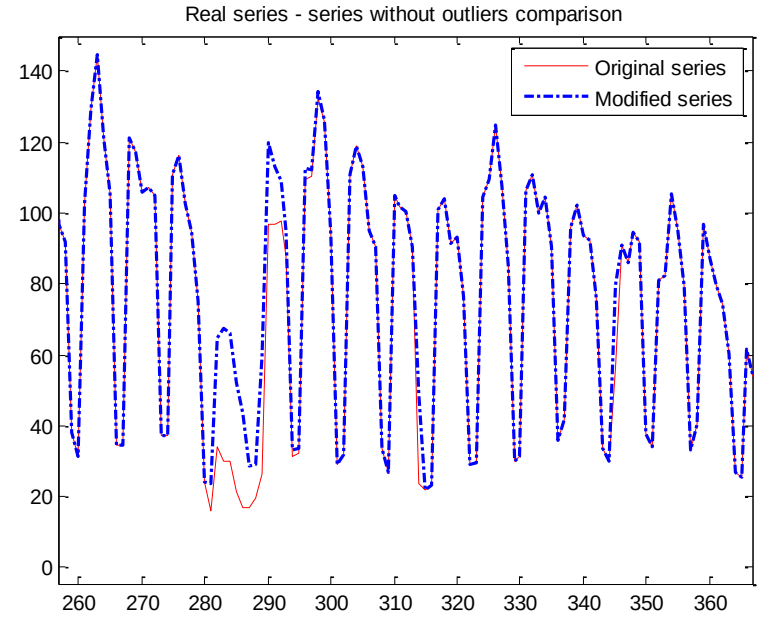


Figure 11: Real series - series without outliers comparison

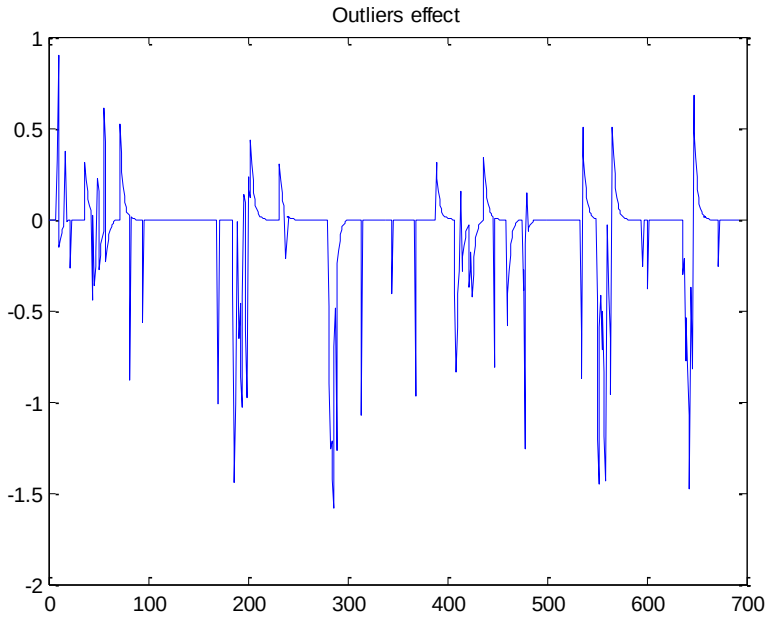


Figure 12: Outlier's impact

4. Neural Network Structure

In this section there is a study and analysis of NN parameters and structure in order to find a net that represents the outlier in this particular time series. The aim of this study is not to find a net that can identify all outliers but to determine that a NN can quantify the impact of atypical situations to automatize its recognition.

4.1. Feature database generation and data analysis

A neural network requires enough features to detect a pattern and work properly. The question of which data to use as features is not as complicated in this particular case due to the fact that time series come from an University Campus. Exams, lesson timetables, local, holidays, weather, etc. are a good example of parameters to take into consideration. All features shall go from 20/06/2014 to 18/05/2016 to match the series range of time.

The weather is a relevant field to extract features from maximum temperature, to precipitations. There are 3 schools in the campus: ETSETB, FIB and ETSECCPB. Their timetables and calendars are used as features, as well as events and parties. Together, and with other features that may be relevant such as local football team matches, they sum up to 67 features for each data point. This set of features is big enough for the purpose of the project.

As it is mentioned in the previous section the NN need to be able to detect if a day is an atypical situation, classify it as an Additive Outlier, Transitory Change or both, and quantify its impact. Some of the outliers are caused by events that do not happen the same day, for example a holiday in the middle on the week can trigger an increment of Internet traffic the next day. Regular Neural Networks have no sorting memory and the events causing the singular behavior are not always in adjacent days. For this reason a NN would not be able to properly detect all outliers. Taking that into account, for a better performance it might be useful to resort to other techniques such as Recurrent NN or LSTM in the future. In this project, a regular NN will give the aimed results, though.

4.2. Neural Networks training and parameter selection

The classification process is divided in three parts: identification of singular situations, classification by type, and quantification. By addressing these steps one by one it is easier to minimize the error in each of them. As a result there are three different NN to train and validate. All three are tested with some training functions and a single hidden layer of several neurons. The most successful training algorithms after a previous analysis are conjugate gradient (CG), Levenberg-Marquadt (LM) and gradient descent with momentum backpropagation (GDM).

4.2.1. Binary identification Neural Network

For this first NN the target is only a binary identification between two classes: regular days and abnormal days. The objective of this net is to cluster all the outliers for them to be analyzed in the next step of the process. For this reason it is more important not to have false negatives than false positives. If an atypical day is not well classified it is lost for the rest of the process, but if a regular day is mistakenly labeled, it can be discarded later. This needs to be taken in consideration when evaluating the error function in the training state. Data is distributed among the training, validation and testing sets with the following relative weights: 80%, 10% and 10% respectively. To avoid errors based on having a set that have few outliers and is not useful for the training state, it is necessary to control the number of elements of each class present in the sets. Depending on how divisions are made they can produce different results due to the random distribution of the elements. For this reason each group of parameters needs to be tested several times varying the sets of vectors. The number of iterations selected for this dataset is 100. There are then 3 parameters to change during the testing: the weight of false negatives, the training algorithm and the number of neurons in the hidden layer. The range selected for the weight is from 5% to 20%. The neurons in the hidden layer go from 50 to 100.

Once the simulation is done with all the possible combinations, the result is measured with the confusion matrix of the combination of each 100 iterations. The one that has the best true positives/false negatives ratio and less global error is the chosen set of parameters. From the 100 iterations of this one, the net with the less error is selected. As it is more important to avoid missing any outlier than a lower total error, the performance shown in table 2 is not the highest. This is caused by a large amount of false positives that will be dismissed on following steps. Table 2 shows a confusion matrix, the center section represents the quantity of true positives, false positives, true negatives and false negatives and their percentage over the total. The right side of the matrix contains the volume of samples classified as true or false by the NN and the success and error percentages. The bottom row indicates the sensitivity and specificity of the NN. Finally, at the bottom-right cell there is the global success and error rate.

		Target Class		
		Regular data	Outlier	
Output Class	Regular data	355 50.79%	0 0%	355 100% 0%
	Outlier	267 38.2%	77 11.02%	344 22.38% 77.62%
		622 57.07% 42.93%	77 100% 0%	699 61.8% 38.2%

Table 2: Confusion matrix of the selected net for binary identification

4.3.2. Outlier classifier

The number of vectors that contain outliers have been reduced from 699 to 344 in the previous section. The NN used in this step is a classifier for 3 classes, Additive Outlier (AO), Transitory Change (TC) or Regular Data (RD). As in the previous NN an error classifying wrong a TC or an AO is worse than assigning a RD as an outlier. During this training process there are taken into account the same restrictions and considerations than in the identification one. There are only 2 training algorithms this time, Levenberg-Marquadt and gradient descent with momentum backpropagation. The three classes are identify as follows: -1 for AO, 0 for RD and 1 for TC. The decision of which is the best combination of parameters and which net is the most appropriate is based on the false positives of RD, and the percentage of true positives of AO and TC with the same weight. This last consideration is taken into account because the number of TC in the series is much lower than the quantity of AO. The result is a net that discards 113 RD and misclassifies 8 outliers, reducing the size of the vector set to 226.

		Target Class			
		AO	RD	TC	
Output Class	AO	38 11.05%	108 34.30%	2 0.58%	148 25.68% 74.32%
	RD	4 1.16%	113 32.85%	1 0.29%	118 95.76% 4.24%
	TC	1 0.29%	62 18.02%	15 4.36%	78 19.23% 80.77%
		43 88.37% 11.63%	285 82.85% 17.15%	18 83.33% 16.66%	344 48.26% 51.74%

Table 3: Confusion matrix of the selected net for outlier classification

4.3.3 Outlier quantification

Quantifying the values of the outliers requires the use of two NN instead of one due that AO and TC elements have different causes and cannot be treated the same at this point. Both Neural Networks work in a similar way than the previous ones. They run 100 times for each set of parameters to minimize the effect random events could have on the result. This time the parameters to test are only the number of neurons in the hidden layer and the training algorithm. There is no need to change the weights because it could modify the result values for the outliers. The range for the hidden neurons goes from 50 to 120 and the training algorithms are the same as the last section.

The objective of these two NN is no longer classifying the elements but giving a measurable value that will be used later in the TS forecasting. The performance is measured with the mean squared error (MSE) of the calculated outliers and the real outliers. The minimum error determines the net that best adjusted the quantification of outliers. Observing the error in figure 13 it can be evaluated as a good result due that the mean value of the series is 62.211 and the biggest error in an outlier is 1.8. The MSE for the best case is 0.0444, which can be consider as quite good for this series.

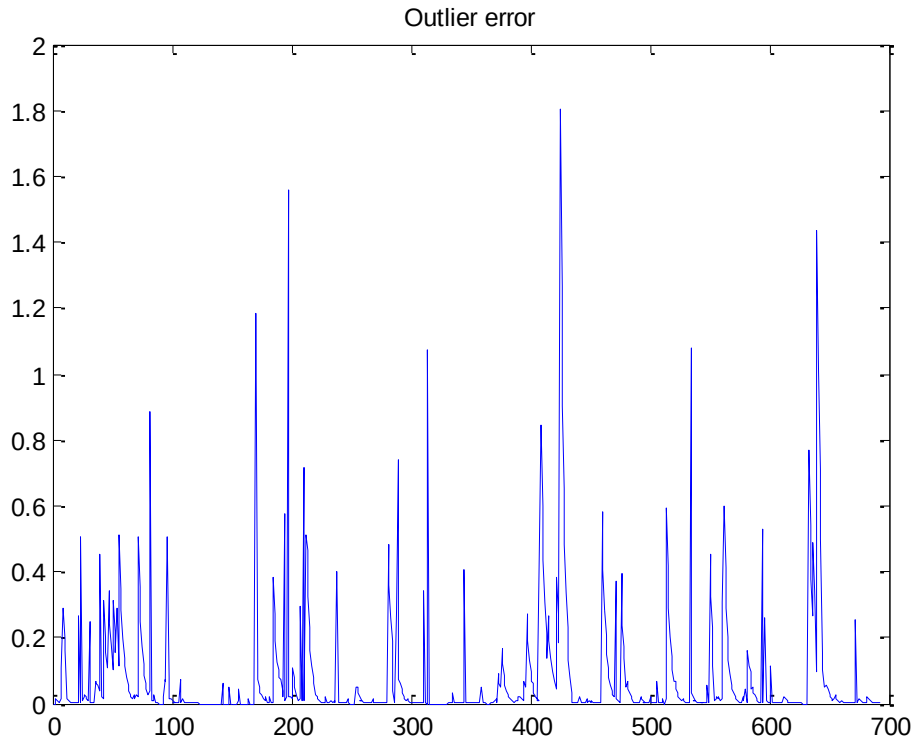


Figure 13: Error between calculated and real outliers

5. Performance and results

This section is aimed to study and evaluate results and analyze the impact in forecasting. In the previous sections an ARIMA model that fits the series had been generated and used to extract the outliers to create a NN structure that is able to classify them and quantify their value. To finalize the process the outliers calculated by the NN are subtracted to the series to fit the model and generate a more clear prediction. In figure 14 there is a sample of both series and it can be appreciated how the outliers are corrected. For example in sample 368 and 389 there are two AO's, a negative and a positive one. A TC correction can also be appreciated in sample 408.

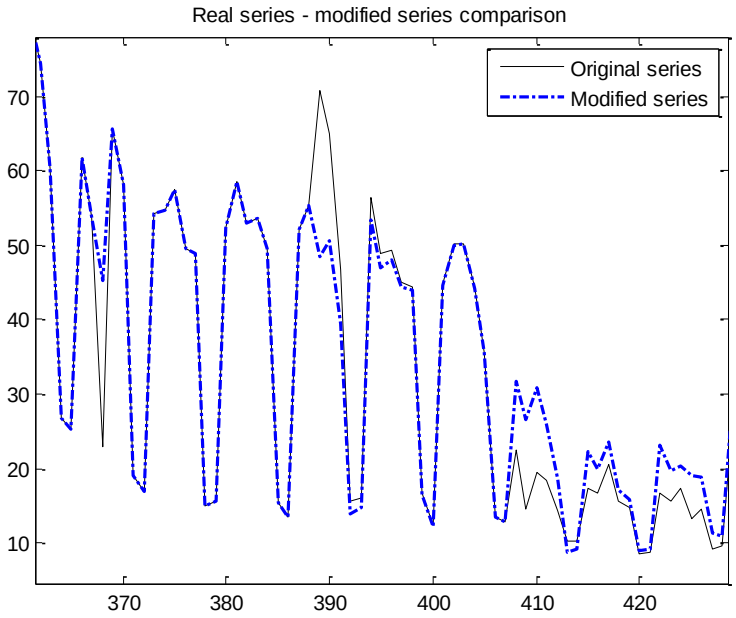


Figure 14: Real Series - Modified Series Comparison

The forecasting is done with this modified series and calculated outliers are added to the prediction to improve its accuracy. The model is tested on different samples with and without outliers in them. In the following table there are a couple of examples with outliers in the forecasted week. The results have been analyzed taking into account two parameters, the mean squared error (MSE) between the prediction and the real series and the separation between the upper and lower condition intervals that shows the accuracy of the outcome. The black line is the real series that determines the expected prediction, the green and red lines are the forecast and the confidence intervals of the modified and original model respectively. In the left figure it can be appreciated that the prediction is deviated from reality in both the regular and the modified model. The modified model, though, represents a more correct representation of the series. The confidence intervals more distanced in the original model and the real series is not even in them. The figure of the right shows that the outlier's modification allow the model to

follow the series shape in a more precise way and the series is kept inside the confidence intervals.

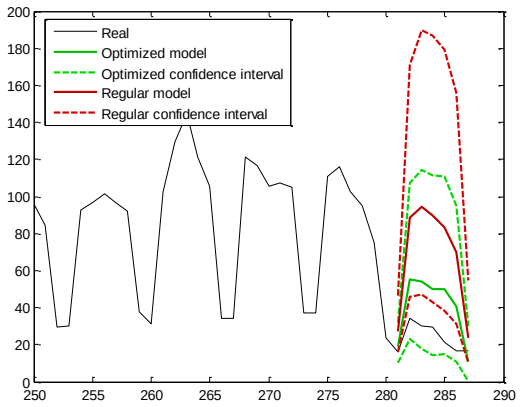
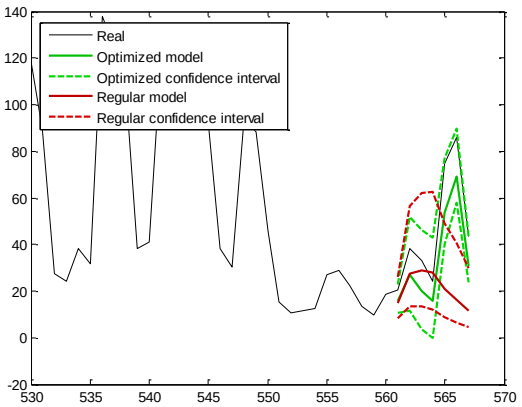
	
MSE improvement of 83.519% Confidence interval improvement of 43.412%	MSE improvement of 85.703% Confidence interval improvement of 47.131%

Table 4: Examples of forecasting with outliers in the prediction

When there is no detected outliers in the prediction the results are pretty similar in the old and new model. In table 5 there are 2 examples of forecasting that show how although there is not a large difference in the prediction, the confidence intervals considerably better in the modified model. That is due the prediction has been done using a series subtracting the calculated outliers.

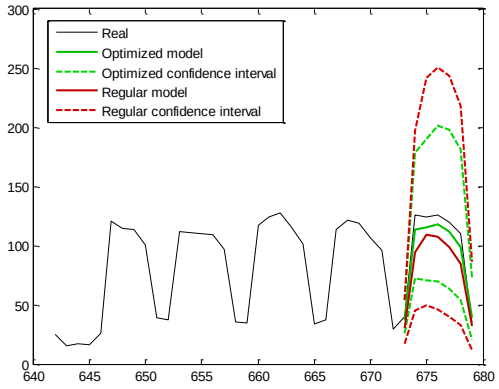
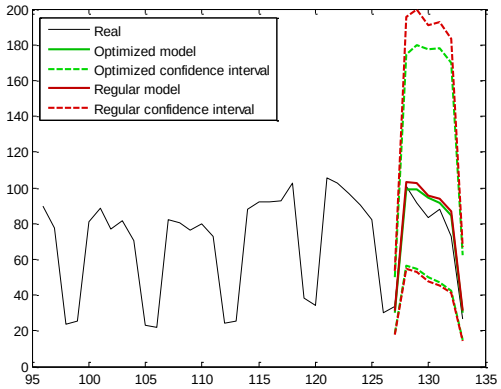
	
MSE improvement of 81.895% Confidence interval improvement of 33.369%	MSE improvement of 32.536% Confidence interval improvement of 12.467%

Table 5: Example of forecasting without outliers in the prediction

The results of the adjusted model are predominantly better than the original model. This can be detected using the cumulative density function (CDF) of the error of both models. In figure 15 there is the normalized CDF of the error of forecast of last 49 weeks of the series, one week each time, for the modified and original model. This graph shows a cumulative function of errors, up to 1 that equals the maximum error. If the model was perfect and there were no deviations, the shape would be equal to a step function. It can be appreciated in this figure that the modified model errors draw a CDF above the original model that shows a higher volume of lower errors.

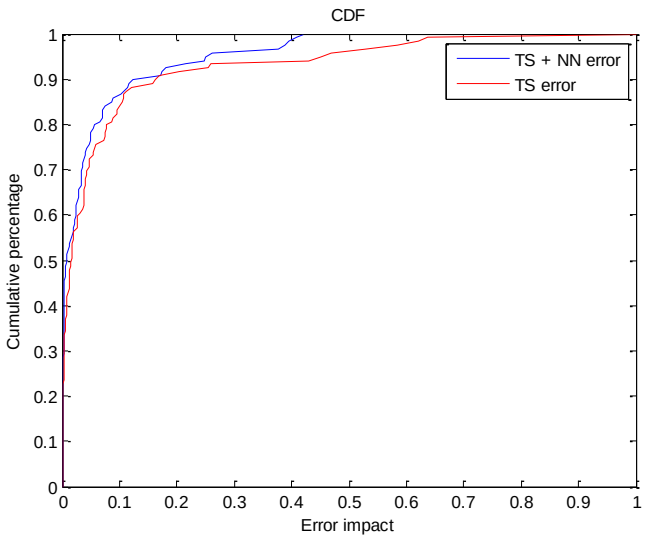


Figure 15: CDF of 49 weeks (1 week long forecasting)

6. Budget

The total budget for this project development is:

Project	Cost
Instrument	103.33 €
Personnel	7260 €
Data	100 €
Total	7463.33 €

Table 6: Project Budget

The cost breakdown for Instruments is:

Instrument	Price(€)	Years	Amortization/hour (€)	User (h)	Cost (€)
Laptop	1800	7	0.0294	720	21.13
Matlab	2000	2	0.1142	720	82.2
				Total	103.33

Table 7: Instrument Budget

The cost breakdown for Personnel is:

Name	Rank	Wage (€/h)	Hours	Cost (€)
Albert Cabellos	Project Director	30	50	1500
Javier González	Junior Engineer	8	720	5760
			Total	7260

Table 8: Personnel Budget

The cost breakdown for Data is:

Data	Price(€)
Procedures for data acquisition	100

Table 9: Bureaucratic Budget

7. Conclusions and future uses

This project purpose is to evince a way to improve forecasting using time series modeling and machine learning techniques with external data.

ARIMA modeling generates a model well fit to the time series, outliers caused by atypical situations generates a substantial error though. This effect in the model causes a deviation in the predictions as seen in the previous sections.

Using a structure of neural networks and a set of external data the system is able to correlate the outliers in the time series with predictable events such as weather, timetables, etc. and compensate the error they generate in the forecast. The results determine that a regular NN is able to correct most of the outliers, but cannot detect the ones caused by adjacent samples. This produces a deviation in the correction that need to be solved by approaching the problem with other ML techniques such as Recursive NN.

The results of the project have been successful. The model based on time series and Artificial Neural Networks presents a decrease of forecast error up to 85% and a considerable improvement of confidence intervals. The system created generates fairly more accurate predictions than a regular ARIMA model.

The results meet the expectations of the project for this particular set of data but they are not the optimal. Data acquisition has been proven to be a struggle because of the particularities of the project. Due to time and labor limitation, the project did not qualify for a more complex dataset. There is a long way to automatize the process for any dataset and polish the techniques used in this project. In the future this techniques shall be applied to larger datasets with more complex trends and outliers.

Bibliography

[1] D. Peña, G. C. Tiao, Ruey S. Tsay, "*A Course in Time Series Analysis*". December 2000.

[2] Dr. Iain Pardoe, Dr. Laura Simon and Dr. Derek Young, "*Regression Methods",* STAT 501. [Online] Available: https://onlinecourses.science.psu.edu/stat501/

[3] Thomas Hill, P. Lewicki, *"Statistics: Methods and Applications",* DELL Inc, 2013. [Online] Available: http://documents.software.dell.com/statistics/current/textbook

[4] Richard A. Davis, "*Introduction to Statistical Analysis of Time Series*", Columbia University. [Online] Available: http://www.stat.columbia.edu/~rdavis/lectures/Session6.pdf

[5] D. Antoni Espasa, Dolores G. Martos, "*Econometría II Grado en finanzas y contabilidad*", U. Complutense Madrid. [Online] Available: http://www.est.uc3m.es/esp/nueva_docencia/comp_col_get/lade/Econometria_II_NOdocencia/Documentaci%C3%B3n%20y%20apuntes/TEMA%206_Metodolog%C3%ADa%20Box-Jenkins.pdf

[6] Robert Nau, "*Statistical forecasting: notes on regression and time series analysis*", Duke University, 1 May 2016. [Online] Available: http://people.duke.edu/~rnau/411home.htm

[7] "*NIST/SEMATECH e-Handbook of Statistical Methods*", 30 October 2013. [Online] Available: http://www.itl.nist.gov/div898/handbook/

[8] Kenjiro Cho, Ryo Kaizaki, "*MAWI Working Group Traffic Archive".* [Online] Available: http://mawi.wide.ad.jp/mawi/

[9] D.F. Specht, "*IEEE Transactions on Neural Networks and Learning Systems",* IEEE Xplore, 6 August 2002.

[10] A.K. Mishra, V.R. Desai, "Drought forecasting using feed-forward recursive neural network", Ecological Modelling Volume 198, 15 September 2006.

Appendix 1

Time Series Modeling

In this appendix the Time Series Modeling methodology used in this project is described on detail.

ARIMA/ARMA Models

There are three main classes in modeling or time series: autoregressive (AR) models, integrated (I) models and moving average (MA) models. All three are based in the idea that the current data point depend only in previous data.

AR models

An autoregressive model *AR(p)* is a mathematical model where a time series is regressed on previous values from that same time series following the next example. Where β are constant parameters and ϵ is the error between the model and reality. This particular equation is a representation of an AR(1) model, for it takes into account only the previous data point.

$$y_t = \beta_0 + \beta_1 y_{t-1} + \epsilon_t$$

MA models

A moving average model *AM(q)* is a mathematical model based in a moving average term which is a past error multiplied by a coefficient. The example below shows an AM(2), where μ is the mean value of the time series.

$$y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2}$$

I models

An integrated model *I(d)* by itself cannot generate a mathematical model of a time series. It is based on the idea that instead of taking into account the data in AR or MA model, it is the different between values that generates de model. This can be applied to reduce the non-stationarity of a time series.

The combination of these three ideas produces two types of more complex classes: *autoregressive moving average models* *ARMA(p,q)* and *autoregressive integrated moving average models* *ARIMA(p,d,q)*. When there are seasonal and unseasonal components in a time series the model must be adjusted to correct the seasonality using lags multiples

of the seasonality (S). This way a data point depend on any S^{th} previous data points. This ARIMA model is defined as two, one for unseasonal components and another for seasonal ones, as follows: ARIMA(p,d,q)(P,D,Q)S.

Box-Jenkins Methodology

Box-Jenkins methodology is an approach to standardize the creation ARIMA models for time series. It is based in a procedure to find the fittest models to a given TS. For this methodology to work it is necessary that the time series is stationary and of mean equal to zero. If it is not the case, there are some methods used to adapt it; such as an integrated model or a previous before differentiation. Box-Jenkins methodology is divided in three processes: Model Identification, Model Estimation and Model Validation.

Model identification

The first steps in the identification process is determine whether or not the series is stationary or has any significant seasonality. The stationarity of a sequence can be determined by plotting the data points directly, but it can also be detected from an autocorrelation plot. This plot is the representation of the autocorrelation between every data point of the TS and represented with the difference or lags between them. A plot with a very slow decay shows the non-stationarity of the sequence.

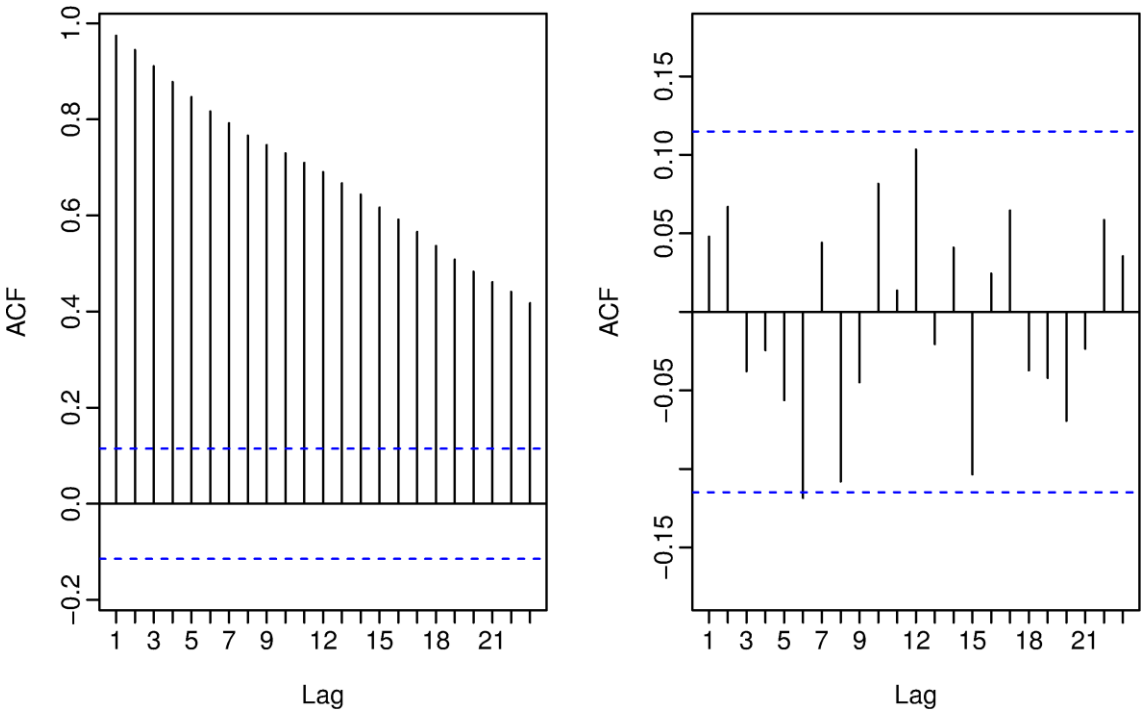


Figure 16: ACF of a stationary sequence (left) and of its differentiation (right)

Determine the seasonal component of a time series can be assessed from an autocorrelation plot, a spectral plot or a seasonal subseries plot. The autocorrelation plot shows in which lags the autocorrelation between data points is stronger. The spectral components of the sequence can determine the periodicity of the TS in the frequency domain. The seasonal subseries can only be done once the period is known and helps understanding the sequence evolution through one season.

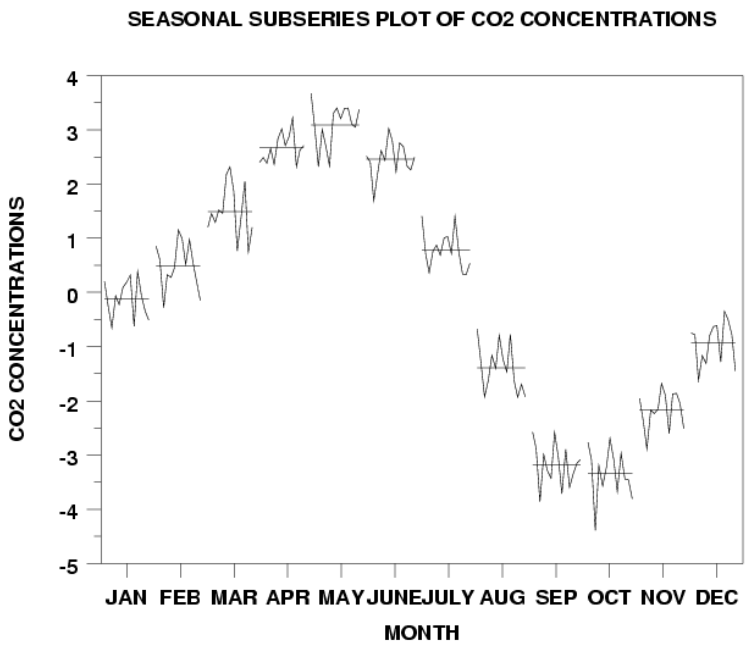


Figure 17: Seasonal subseries plot

For a time series stationary there are multiple solutions. The non-stationarity can be achieved by differencing the sequence until it loses its stationarity, fitting a curve and subtracting the fitted values, etc. The seasonality is treated different, it is not removed from the series but taken into account in the ARIMA model as a seasonal component.

Identify the parameters for the AR and MA models (p,q) are the next step in this process. It is done by using the autocorrelation plot and partial autocorrelation plot. The MA degree is detected by the biggest lag that overpass a threshold located at $\pm 2/\sqrt{N}$. The AR degree is determined by the waveform of the autocorrelation function. An exponentially decreasing shape indicates a first degree AR, a higher degree implies exponential and sigmoid mixes.

Appendix 2

Outliers detection

In this section the methodology used to detect outliers in an ARIMA model is explained. There are three main types of outliers, Additive Outliers (AO), Transitory Changes (TC) and Level Shifts (LS). The process required to extract them from a Time Series Model is transforming the ARIMA pattern to an Autoregressive one.

To simplify the explanation of this process the conversion MA(1) to AR(∞) is taken as an example. The equation 1 is the expression for a MA(1) model with mean value 0. It can be written as equation 2 were L is the delay of the samples.

$$y_t = \epsilon_t - \theta\epsilon_{t-1}$$

Equation 1

$$y_t = \epsilon_t - \theta L\epsilon_t$$

Equation 2

The desired result would be to obtain y_t as an infinite sum dependent of y previous values from y_{t-1} to y_0. It is possible to isolate ϵ_t and generate equation 3 based on the geometric series formula for $|k| < 1$.

$$\epsilon_t = \frac{y_t}{1-\theta L}$$

$$\sum_{k=0}^{\infty} ar^k = \frac{a}{1-r}$$

$$\epsilon_t = \sum_{k=0}^{\infty} y_t(\theta L)^k = y_t + y_t\theta L + y_t(\theta L)^2 + \cdots = y_t + y_{t-1}\theta + y_{t-2}\theta^2 + \cdots$$

Equation 3

Using this last equation it is possible to determine the AR model that allows the outliers detection for this time series, as shown in the equation below.

$$z_t = y_t = -\sum_{k=1}^{\infty} y_t(\theta L)^k + \epsilon_t = \epsilon_t - y_{t-1}\theta - y_{t-2}\theta^2 - \cdots$$

Equation 4

The procedure to determine the outliers is to check for unusual high values in the residues that indicates an outlier, and find the corresponding outlier type that best fits them. The impact of each outlier is determined by the ARIMA model and the characteristics of each of them.

Additive Outliers

These outliers affect only one sample and generate an impact in the p following data points for an AR(p) model. Taking into account a_t as the correct residue of a time series, the model for calculating the residues with an outlier in t = h is as follows in equation 5. Where ω_A is the impact of the outliers, I_t^h is 0 when t ≠ h and 1 otherwise and $\phi(B)$ is AR operator of p degree.

$$e_t = a_t + \omega_A I_t^h - \phi_1 \omega_A I_{t-1}^h - \cdots - \phi_p \omega_A I_{t-p}^h$$

Equation 5

Transitory Changes

The impact of these outliers in the series is similar to the AO effect but they have a transitory component that decreases in time. Their value is quantified with the equation $\omega/(1-0.7B)$. The residues function for a TC in t = h is shown in equation 6 and depends on the weight of the impact and the AR degree. Where $S_t^h = 1$ for t ≥ h and 0 otherwise.

$$e_t = a_t + \frac{\omega_{TC}}{1-0.7B} S_t^h - \phi_1 \frac{\omega_{TC}}{1-0.7B} S_{t-1}^h - \cdots - \phi_p \frac{\omega_{TC}}{1-0.7B} S_{t-p}^h$$

Equation 6

Level Shifts

Level Shifts cause an additive value to all the samples where t ≥ h. Are the least common due that are caused by exceptional permanent situations. Their interpretation is similar to a TC but with a degradation of 0. The residues equation is described in equation 7 for a LS at t = h, where $R_t^h = t + 1 - h$ for t ≥ h and 0 otherwise.

$$e_t = a_t + \omega_R R_t^h - \phi_1 \omega_R R_{t-1}^h - \cdots - \phi_p \omega_R R_{t-p}^h$$

Equation 7

Appendix 3

Forecasting with ARIMA models

The methodology used to forecast samples is based on obtain predictions from the previous data points using an infinite AR model and calculate the confidence intervals with an infinite Ma model.

Obtain predictions from an AR(∞)

The transformation ARMA to AR can be found in Appendix 3. In this section it is explained how to use the model to obtain the desired forecast. An AR model is based on represent the values as a linear combination of previous samples. This feature makes the forecast process automatic, generating the next data point depending only on already existent data. As shown in equation 8, employing the coefficients used to model the time series, a prediction of the pattern is developed; being T the number of samples of the time series. Where I_T is the information known when t=T and $\epsilon_h = 0$ for h > T.

$$y_{T+1|I_T} = E(y_{T+1}|I_T) = E\left(\mu + \theta_1 y_T + \theta_2 y_{T-1} + \cdots + \theta_p y_{T-p+1} + \varepsilon_{T+1}\middle|I_T\right)$$

$$y_{T+1|I_T} = \mu + \theta_1 E(y_T|I_T) + \theta_2 E(y_{T-1}|I_T) + \cdots + \theta_p E(y_{T-p+1}|I_T) + E(\varepsilon_{T+1}|I_T)$$

$$y_{T+1|I_T} = \mu + \theta_1 y_T + \theta_2 y_{T-1} + \cdots + \theta_p y_{T-p+1}$$

Equation 8

Obtain confidence intervals from an MA(∞)

A transformation ARMA to MA follows a similar process that ARMA to AR. An example of AR(1) to MA(∞) is used to simplify the explanation. The equation 9 is the expression for an AR(1).

$$y_t = \mu + \theta y_{t-1} + \epsilon_t$$

$$y_t = \mu + \theta L y_t + \epsilon_t$$

Equation 9

Taking into account a series of mean value equal to 0 and isolating y_t an equation similar to a geometric series is found. Developing the function a MA(∞) is obtained, as shown in equation 10.

$$y_t = \frac{\varepsilon_t}{1-\theta L}$$

$$\sum_{k=0}^{\infty} ar^k = \frac{a}{1-r}$$

$$y_t = \sum_{k=0}^{\infty} \varepsilon_t(\theta L)^k = \varepsilon_t + \varepsilon_t\theta L + \varepsilon_t(\theta L)^2 + \cdots = \varepsilon_t + \varepsilon_{t-1}\theta + \varepsilon_{t-2}\theta^2 + \cdots$$

Equation 10

The variance of the prediction is calculated as the variance conditioned with the information until T. The equation 11 explains the calculations followed to obtain the variance used to generate the confidence intervals.

$$C_1 = V(y_{T+1}|I_T) = E\left(\left(y_{T+1} - y_{(T+1|I_T)}\right)^2\right)$$

$$C_1 = E\left(\left(y_{T+1} - y_{(T+1|I_T)}\right)^2\right) = V(\varepsilon_{T+1}) = \sigma^2$$

$$C_2 = E\left(\left(y_{T+2} - y_{(T+2|I_T)}\right)^2\right) = V(\varepsilon_{T+2} + \theta\varepsilon_{T+1}) = (1+\theta^2)\sigma^2$$

$$C_k = E\left(\left(y_{T+k} - y_{(T+k|I_T)}\right)^2\right) = V(\varepsilon_{T+k} + \theta\varepsilon_{T+k-1} + \theta^2\varepsilon_{T+k-2} + \cdots + \theta^{k-1}\varepsilon_{T+1})$$

$$C_k = (1 + \theta^2 + \theta^4 + \cdots + \theta^{2k})\sigma^2$$

Equation 11

Using the coefficients of the MA(∞), the variance of any of the k samples can be calculated. The confidence intervals are situated within the 95% probability rate. That means that they are at 1.96 times the standard deviation from the predicted point.